

# DETECTION OF ARP SPOOFING

## PHASE 1 REPORT

*Submitted by*

**Shahasiya Muhammed AC (RCAS2021MCS208)**

*in partial fulfillment for the award of the degree of*

**MASTER OF SCIENCE  
SPECIALIZATION IN  
INFORMATION SECURITY AND CYBER FORENSICS**



**DEPARTMENT OF COMPUTER SCIENCE  
RATHINAM COLLEGE OF ARTS AND SCIENCE  
(AUTONOMOUS)  
COIMBATORE - 641021 (INDIA)  
DECEMBER-2022**

**RATHINAM COLLEGE OF ARTS AND SCIENCE**  
**(AUTONOMOUS)**  
COIMBATORE - 641021



**BONAFIDE CERTIFICATE**

This is to certify that the thesis entitled **Detection of ARP Spoofing** submitted by **Shahasiya Muhammed AC,,** for the award of the Degree of Master in Computer Science specialization in **“INFORMATION SECURITY AND CYBER FORENSICS”** is a bonafide record of the work carried out by him/her under my guidance and supervision at Rathinam College of Arts and Science, Coimbatore

**Mr. Saravana Kumar**  
Supervisor

**Mr.P.Sivaprakash**  
Mentor

Submitted for the University Examination held on 02.12.2022

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

**RATHINAM COLLEGE OF ARTS AND SCIENCE**  
**(AUTONOMOUS)**  
COIMBATORE - 641021

**DECLARATION**

I, **Shahasiya Muhammed AC**, hereby declare that this thesis entitled "**Detection of ARP Spoofing**", is the record of the original work done by us under the guidance of **Mr. Saravana Kumar**, Faculty Rathinam college of arts and science, Coimbatore. To the best of my knowledge this work has not formed the basis for the award of any degree/diploma/ associateship/fellowship/or a similar award to any candidate in any University.

**Signature of the Students:**

Shahasiya Muhammed AC

**Place: Coimbatore**

**Date: 02.12.2022**

**COUNTERSIGNED**

Mr. Saravana Kumar  
Supervisor

# Contents

<b>Acknowledgement</b>	<b>iii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Abbreviations</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Objective of the project . . . . .	3
1.2 Scope of the Project . . . . .	3
1.3 Module Description . . . . .	4
1.4 Existing System . . . . .	5
<b>2 Literature Survey</b>	<b>7</b>
2.1 SDN paradigm: . . . . .	7
2.2 Modified ICMP and Voting: . . . . .	8
<b>3 Methodology</b>	<b>10</b>

3.1	Implementaion and working process: . . . . .	10
3.2	Software and Hardware Requirements: . . . . .	12
3.3	Technologies Used: . . . . .	12
3.4	System Design . . . . .	14
<b>4</b>	<b>Experimental Setup</b>	<b>16</b>
4.1	Create Intent . . . . .	16
4.1.1	Get user Input . . . . .	16
4.1.2	Get User Response . . . . .	18
4.2	User Interface . . . . .	18
<b>5</b>	<b>Conclusion</b>	<b>20</b>
5.1	Future Works . . . . .	21
	<b>References</b>	<b>22</b>

## Acknowledgement

On successful completion for project look back to thank who made in possible. First and foremost, thank “**THE ALMIGHTY**” for this blessing on us without which we could have not successfully our project. We are extremely grateful to **Dr.Madan.A. Sendhil, M.S., Ph.D.**, Chairman, Rathinam Group of Institutions, Coimbatore and **Dr. R.Manickam MCA., M.Phil., Ph.D.**, Secretary, Rathinam Group of Institutions, Coimbatore for giving me opportunity to study in this college. We are extremely grateful to **Dr.R.Muralidharan, M.Sc., M.Phil., M.C.A., Ph.D.**, Principal Rathinam College of Arts and Science(Autonomous), Coimbatore. Extend deep sense of valuation to **Mr.A.Uthiramoorthy, M.C.A., M.Phil., (Ph.D)**, Rathinam College of Arts and Science (Autonomous) who has permitted to undergo the project.

Unequally I thank **Mr.P.Sivaprakash, M.E., (Ph.D)**., Mentor and **Dr.Mohamed Mallick, M.E., Ph.D.**, Project Coordinator, and all the Faculty members of the Department - iNurture Education Solution pvt ltd for their constructive suggestions, advice during the course of study. We convey special thanks, to the supervisor **Mr.Saravana Kumar.**, who offered their inestimable support, guidance, valuable suggestion, motivations, helps given for the completion of the project.

We dedicated sincere respect to my parents for their moral motivation in completing the project.

# List of Figures

3.1	Overview of Proposed system . . . . .	15
4.1	Connecting two machine . . . . .	17
4.2	User interface . . . . .	19

# List of Abbreviations

ARP	Address Resolution Protocol
LAN	Local Area Network
IP	Internet protocol
MAC	Media Access Control
RAM	Random Access Memory
NAT	Network Address Translation
TCP	Transmission control Protocol
SDN	Software Design Networks



# Abstract

Address Resolution Protocol (ARP) spoofing is a type of cyber-attack carried out over a local area network (LAN) that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its Internet Protocol (IP) to Media Access Control (MAC) address table. ARP protocol translates IP addresses into MAC addresses because the ARP protocol was designed purely for efficiency and not for security. As long as the attacker is directly connected to the target LAN or in control of a workstation within it, ARP spoofing attacks are very simple to execute. The attack is carried out by an attacker deceiving the default network gateway into believing that their MAC address should be linked to the target's IP address (and vice versa, making the target's MAC address linked to the attacker's IP address). All of the target's traffic to any other device on the network travels through the attacker's computer after the default gateway receives this message and broadcasts its modifications to all other devices on the network, enabling the attacker to inspect or alter it before forwarding it to its actual destination. Since ARP spoofing attacks occur on such a low level, users targeted by ARP spoofing rarely realize that their traffic is being inspected or modified besides man-in-the-middle attack. ARP spoofing can be used to cause a

denial-of-service condition over a LAN by simply intercepting or dropping and not forwarding the target's packets. The ARP spoofing attack can be prevented by this email alert notification which is sent to the administrator, so that the user can be able to know that the attack is happening and the communication can be terminated and the user can stop the attack from happening.

# Chapter 1

## Introduction

Address Resolution Protocol (ARP) spoofing is a type of cyber-attack carried out over a local area network (LAN) that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its Internet Protocol (IP) to Media Access Control (MAC) address table. ARP protocol translates IP addresses into MAC addresses because the ARP protocol was designed purely for efficiency and not for security. As long as the attacker is directly connected to the target LAN or in control of a workstation within it, ARP spoofing attacks are very simple to execute. The attack is carried out by an attacker deceiving the default network gateway into believing that their MAC address should be linked to the target's IP address (and vice versa, making the target's MAC address linked to the attacker's IP address). Once the default gateway has received this message and broadcasts its changes to all other devices on the network, all of the target's traffic to any other device on the network travels through the attacker's computer, allowing the attacker to inspect or modify it before forwarding it to its real destination. Since ARP spoofing attacks occur on such a low level, users targeted by ARP spoofing rarely realize that their traffic is being

inspected or modified besides man-in-the-middle attack. ARP spoofing can be used to cause a denial-of-service condition over a LAN by simply intercepting or dropping and not forwarding the target's packets. Most of the organizations implement LAN for their communication and networking needs. The MAC address is the communication identifier in LANs. So, in order to transport packets and communicate within a LAN, IP addresses must be resolved to MAC addresses. The Address Resolution Protocol handles this resolution (ARP). This protocol, however, has a serious security flaw. The ARP protocol has no state. It doesn't verify whether a request was ever made for the response that was given. As a result, it is vulnerable to an exploit called an ARP Spoofing attack. As long as the attacker is directly connected to the target LAN or in control of a workstation within it, ARP spoofing attacks are very simple to execute. The attack is carried out by an attacker deceiving the default network gateway into believing that their MAC address should be linked to the target's IP address (and vice versa, making the target's MAC address linked to the attacker's IP address). The ARP spoofing attack can be prevented by this email alert notification which is sent to the administrator, so that the user can be able to know that the attack is happening and the communication can be terminated and the user can stop the attack from happening.

## 1.1 Objective of the project

The main objective the project is to address two issues; first, is to identify the attacker and second, is to prevent the ARP Spoofing attack. The attacker will spoof the MAC address to perform ARP spoofing. The goal here would be to check for spoofed MAC address and to prevent the attack by notifying the administrator via terminal and the email

For their networking and communication requirements, the majority of enterprises use LAN. The MAC address is the communication identifier in LANs. So, in order to transport packets and communicate within a LAN, IP addresses must be resolved to MAC addresses. The Address Resolution Protocol handles this resolution (ARP). This protocol, however, has a serious security flaw. The ARP protocol has no state. It doesn't verify whether a request was ever made for the response that was given. As a result, it is more vulnerable to an exploit called an ARP spoofing attack.

## 1.2 Scope of the Project

This project is being developed with the aim to demonstrate and detect ARP Spoofing. To perform Man-in-the-middle (MITM) attacks, ARP spoofing is mostly used. Being able to witness few ARP Spoofing cases, was the principal reason of inspiration to find a solution. The plan is to prepare a script to detect an ARP Spoofing. The outcome of this project.

Already different methods have been proposed to transfer data securely over the

internet. However, none of them has provided a feasible solution to counter the ARP spoofing. Thanks to tools that can be downloaded from the Internet, intercepting the communication between two hosts on a LAN has become quite simple. These instruments employ the Address Resolution Protocol (ARP) spoofing method, which depends on hosts caching reply messages even though the corresponding requests weren't ever sent.

False positives, false negatives, and continual maintenance are potential drawbacks of this strategy. Since ARP spoofing can take many different forms, detection could result in a high number of false alarms, which might cause incident response teams to ignore or suppress alarms without conducting a thorough investigation.

### 1.3 Module Description

**Network Traffic Interceptor:** Network traffic is the amount of data, which moves across a network during any given time. Network traffic is the main component for the network traffic measurement, network traffic control and simulation.

**ARP Sniffer:** A sniffer is a software or hardware tool that allows the user to “sniff” or monitor the internet or network traffic realtime, capturing all the data flowing to and from the user’s computer.

**ARP Spoof Detector:** All devices are uniquely identified by a Media Access Control (MAC) address. Legitimate MAC addresses can easily be spoofed and used for various attacks such as MITM and DOS attacks. In ARP Spoofing attacks, MAC Address

plays an important role.

**Alert Administrator:** Email alert/notification is an email sent to inform the administrator about any spoofed or updates in the ARP spoofing attacks

## 1.4 Existing System

### **SDN paradigm:**

Software Defined Network Authors: : Mohammad Z. Masoud, Yousf Jaradat and Ismael Jannoud

Year: 2015 a new algorithm has been proposed and implemented to prevent this attack and an experiment has been conducted to evaluate the proposed algorithm. The result demonstrated that the new algorithm prevented ARP spoofing and many breaches and attacks utilizing ARP spoofing

The centrality found in SDN so that any application will be deployed in the controller and that will affect all users of the network. The centrality found in SDN so that any application will be deployed in the controller and that will affect all users of the network.

### **Modified ICMP and Voting:**

Authors: Prerna Arote; Karam Veer Arya. Year: 2015. Description: The basic idea of presented architecture is to design such structure on more than three systems, for transmitting ARP and ICMP packets, or the central server plays an important role in proposed scheme. During the failure of any one of the systems, other system can work alone. Therefore, backward compatibility is achieved with the original ARP structure.

### **ASD: ARP Spoofing Detector using OpenWrt:**

Year:2022 . Description:Three information tables, AssocList, ARP cache table, and DHCP table, which are frequently managed by the access point based on a Linux system, are used by the access point-based ARP Spoofing Detector (ASD) to distinguish between ARP spoofing and connections from VM guests.

### **ARP Poisoning Detection and Prevention using Scapy:**

Year:2021. Description: A Python programme employing the Scapy module was used to create the ARP Spoof attack tool. For detecting the above-generated ARP Poisoning attack (or any ARP Poisoning attack in general), a detection algorithm has been proposed, and the algorithm has been implemented using a python script with the scapy library.



# Chapter 2

## Literature Survey

The “ARP spoofing” attack is based on impersonating a system in the network, making the two ends of a communication believe that the other end is the attacker’s system, intercepting the traffic interchanged. Numerous strategies have already been put out for safely transferring data over the internet. But none of them has offered a workable defence against ARP spoofing. Thanks to tools that can be obtained from the Internet, intercepting the traffic between two hosts on a LAN has become relatively simple. These instruments make use of the Address Resolution Protocol (ARP) spoofing technique, which depends on hosts caching response messages even though the corresponding requests weren’t sent.

### 2.1 SDN paradigm:

SDN has been utilized to alleviate and eliminate the problem of ARP poisoning attack. Many other network attacks, including man-in-the-middle, denial-of-service, and session hijacking, have this attack as their foundation. Software Defined Network (SDN) is an emerging network paradigm that introduces a centralized controller to the network.

This controller has the ability to manipulate and program the algorithms and protocols that run over the underlying hardware.

The ability of the SDN approach is to prevent ARP spoofing attack without modifying the original ARP protocol. The centrality found in SDN so that any application will be deployed in the controller and that will affect all users of the network. If the network's users obtain their IP addresses from a DHCP server, SDN controller can inspect DHCP reply messages to obtain the gateway IP address sent to the users in the option field of the message. The main drawbacks of this SDN paradigm was they were focused only on IPv4 and To the SDN structure, No new field will be added to the ARP messages.

## **2.2 Modified ICMP and Voting:**

The basic idea of presented architecture is to design such structure on more than three systems, for transmitting ARP and ICMP packets, or the central server plays an important role in proposed scheme. During the failure of any one of the systems, other system can work alone. Therefore, backward compatibility is achieved with the original ARP structure

The main features of modified ICMP and Voting are it is less costly and easy to use but the main drawback of this system was To maintain fairness among different nodes where computational power of machines is more diverse. The algorithm and the technology used were router/switch and ARP packets

Possible weaknesses in this approach include false positives, false negatives, and ongoing maintenance. Since ARP spoofing can present itself in many forms, detection

could cause excessive false alarms, which could lead to incident response teams ignoring or suppressing alarms without proper investigation. From the literature review, analysis were done and the study revealed various methods used to detect and prevent ARP Spoofing attacks. There are many advantages as well as disadvantages to the above mentioned researches. To mitigate the drawbacks present in existing systems the solution proposed is to create a script, which could effectively detect the ARP Spoofing in realtime scenario. This will help in mitigation of false alerts/alarms, and provide better results to the organization and in turn help maintain their security posture with respect to the ARP Spoofing attacks

# Chapter 3

## Methodology

### 3.1 Implementaion and working process:

This project can be implemented with a real network environment consisting of three systems (attacker, victim and gateway). VMware workstation pro is setup using NAT gateway where both the systems have Kali Linux distribution installed. All the systems are connected over wired network. The attacker's system is configured to have 1 GB RAM, 60 GB Harddisk and a processor whereas the victim's system have 2 GB RAM, 60 GB Harddisk and a processor.

The UI used for this project is the command-line interface. A command-line interface (CLI) is a text-based user interface (UI) used to run programs, manage computer files and interact with the computer. CLIs accept as input commands that are entered by keyboard; the computer then runs the commands invoked at the command prompt. The project is designed in such a way that the UI and the other elements will be user friendly. The programming languages used for this project is Python. A command line shell, Bash is used to run the scripts. The benefits of using shell is that it enables users

to write programs that can be run on the system from the command line, direct the output of one program to be the input for another program, and system variables which can be set at the command line, or the values of those variables displayed.

### **Algorithm Implementation**

The current algorithm deals with 3 systems Victim, Attacker, and Gateway over the network connected through the wired connection. Attacker uses arpspoof tools to play the role of man in the middle like an intruder. Algorithm checks the ARP cache. With the help of Arpspoof tool, ARP poisoning is done between Victim and Gateway using arpspoof command. Now all the traffic is routed to an Attacker. It captures all kind of traffic including arp request and arp response packets flowing among hosts. The algorithm detects the spoof and alerts the administrator. Additionally, the network and ARP traffic is monitored and logged to a log file which will be saved in the server (here, victim).

- \* Programming Language: Python

- \* Scripting Language: Bash

- \* Tools Used: Tshark, Tcpdump, Wireshark, Arpspoof

## 3.2 Software and Hardware Requirements:

### Hardware Software Requirements:

- \* Laptop with 4 GB RAM
- \* Preferred Linux distro: Kali Linux 2021.4
- \* Python Version: 3.9.0
- \* A Router

## 3.3 Technologies Used:

### Scapy:

Scapy is a library supported by both Python2 and Python3. It is used for interacting with the packets on the network. It includes a number of features that make it simple to counterfeit and tamper with the packet. Scapy's module allows us to build a variety of network tools, including packet dumpers, network scanners, and ARP spoofers. Using this module, more complex tools for network security and ethical hacking can be produced. **Netifaces:**

It is a portable third-party library in Python to enumerate network interfaces on local machine. Historically it has been difficult to straightforwardly get the network addresses of the machine on which the Python scripts are running without compromising the portability of your script. Netifaces takes care of enumerating interfaces, network addresses and preserves the portability.

### **TCP dump:**

Tcp dump is a data-network packet analyzer computer program that runs under a command line interface. It enables the user to view TCP/IP and other packets that are sent across a network to which the computer is connected, as well as other packets. Tcpdump is free software that is distributed under the BSD licence. Most Unix-like operating systems support Tcpdump. Tcpdump utilises the libpcap library in those systems to collect packets. WinDump, a port of tcpdump for Windows, makes use of WinPcap, the Windows implementation of libpcap.

### **Tshark:**

TShark is a network protocol analyzer. It is the terminal version of Wireshark that supports similar options and is more useful. It lets the user capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. TShark's native capture file format is pcapng format, which is also the format used by Wireshark and various other tools. This CLI is more easy to use than GUI.

### **Proposed Algorithm:**

1. Start
2. Fire up the script
3. Import necessary packages and modules
4. Intercept Network Traffic
5. Filter ARP Traffic
6. Select an interface

7. Monitor the selected interface
8. If ARP Spoof attack detected, print "Attack Detected"
9. Alert the administration via email.
10. Go back to 6
11. If "Ctrl + C" is pressed, end the script.
12. End

### **3.4 System Design**

Network traffic is the amount of data, which moves across a network during any given time. Network traffic is the main component for the network traffic measurement, network traffic control and simulation. A sniffer is a software or hardware tool that allows the user to "sniff" or monitor the internet or network traffic realtime, capturing all the data flowing to and from the user's computer. All devices are uniquely identified by a Media Access Control (MAC) address. Legitimate MAC addresses can easily be spoofed and used for various attacks such as MITM and DOS attacks. In ARP Spoofing attacks, MAC Address plays an important role. Email alert/notification is an email sent to inform the administrator about any spoofed or updates in the ARP spoofing attacks.



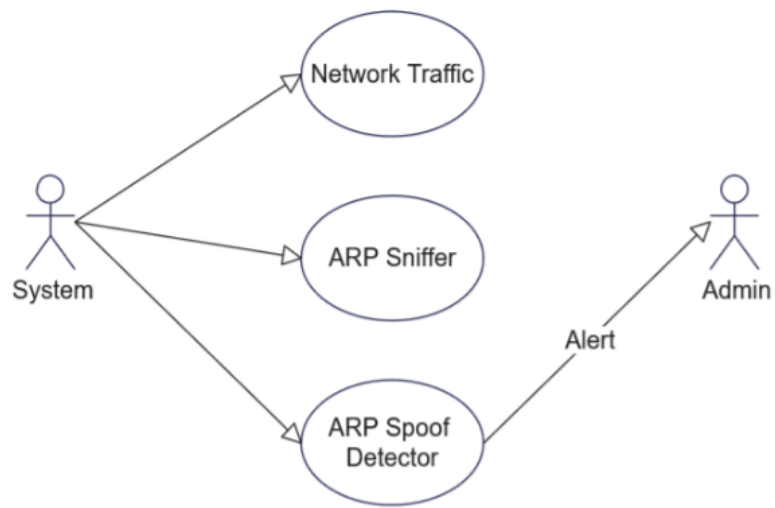


Figure 3.1: Overview of Proposed system

# Chapter 4

## Experimental Setup

### 4.1 Create Intent

Two Linux-powered workstations will be networked together. The ARP database will be updated with its MAC address and IP address after pinging the two networks.

#### 4.1.1 Get user Input

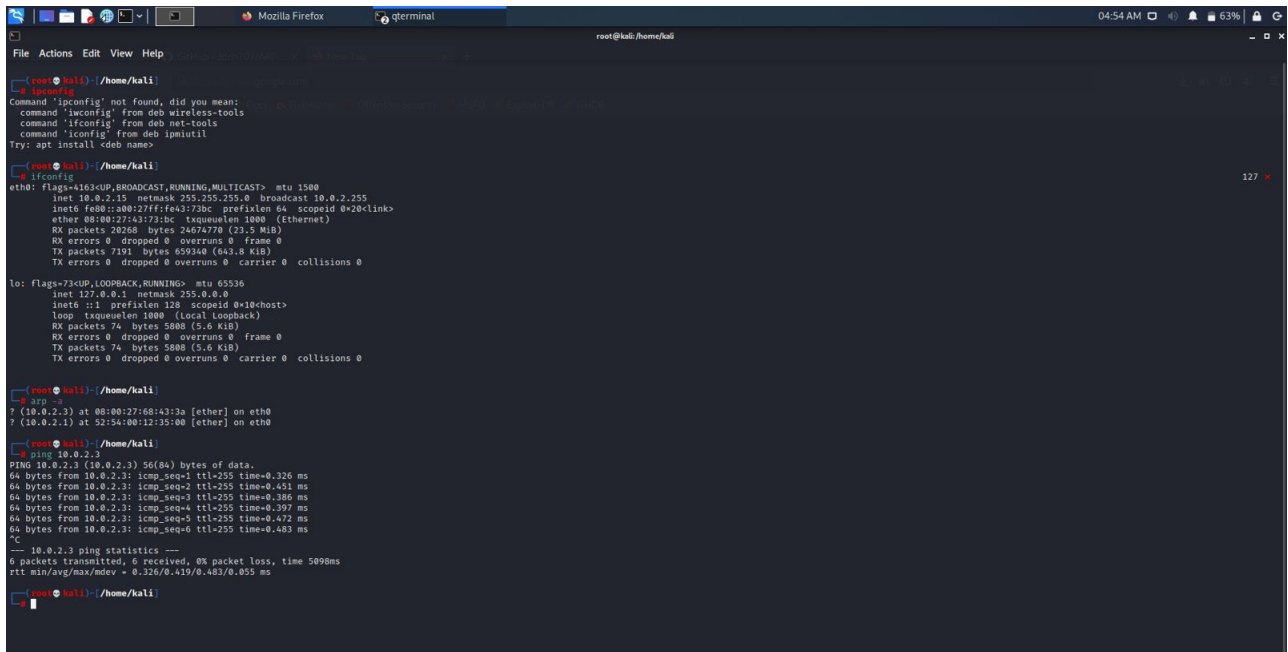
In this approach, two linux machines have created separately and those machines need to be assigned in same network which is NAT network. After that two machines need to be pinged using the ping command. After pinging the two machines each other, now the MAC address will be stored in ARP table. This ARP table will show the all previously connected machines. So by observing the MAC address in the ARP table, we can come into point that two machines are connected each other

#### **Commands:**

\* **ifconfig** - To check the IP address and the MAC address of the particular system and the interfaces which the machine is connected.

\* **ping**- Command to start communication between the two machines.

\* **arp -a** Command to check the ARP table to display the MAC address of the machines which connected.



```
root@kali: /home/kali
# ifconfig
Command 'ifconfig' not found, did you mean:
  command 'iwconfig' from deb wireless-tools
  command 'ifconfig' from deb net-tools
  command 'ifconfig' from deb ipmiutil
Try: apt install <deb name>

root@kali: /home/kali
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe43:73bc prefixlen 64 scopeid 0<2<link>
    ether 82:00:27:a3:72:bc txqueuelen 1000 (ethernet)
    RX packets 20268 bytes 24674770 (23.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7191 bytes 659240 (643.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<1<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 74 bytes 5800 (5.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 74 bytes 5800 (5.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali: /home/kali
# arp -a
? (10.0.2.3) at 00:00:27:60:43:2a [ether] on eth0
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on eth0

root@kali: /home/kali
# ping 10.0.2.3
PING 10.0.2.3 (10.0.2.3) 56(84) bytes of data:
64 bytes from 10.0.2.3: icmp_seq=1 ttl=255 time=0.326 ms
64 bytes from 10.0.2.3: icmp_seq=2 ttl=255 time=0.451 ms
64 bytes from 10.0.2.3: icmp_seq=3 ttl=255 time=0.386 ms
64 bytes from 10.0.2.3: icmp_seq=4 ttl=255 time=0.397 ms
64 bytes from 10.0.2.3: icmp_seq=5 ttl=255 time=0.472 ms
64 bytes from 10.0.2.3: icmp_seq=6 ttl=255 time=0.483 ms
^C
--- 10.0.2.3 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5098ms
rtt min/avg/max/mdev = 0.326/0.419/0.483/0.055 ms

root@kali: /home/kali
```

Figure 4.1: Connecting two machine

### 4.1.2 Get User Response

When the user has given the input for checking the ARP table it will show the MAC addresses in the ARP table and IP addresses. The user will be using the Command Line Interface. The script could be run using a terminal or command prompt.

## 4.2 User Interface

The user interface's design rules for the project are as follows:

- \* In order to improve usability, an interface needs to be well designed to be "user-friendly".
- \* Use simple interfaces
- \* Have extensive documentation
- \* Good color scheme
- \* User friendly
- \* Easy to use
- \* Simple and understandable

**User Interfaces** The user will be using the Command Line Interface. The script could be run using a terminal or command prompt.

**Hardware Interfaces** The user can use any router and its dashboard for this project. There is no limitations as such.

**Software Interface** The Operating system used for this project is Linux and the programming language is Python. The tools that will be used are Wireshark, TShark

<b>Component</b>	<b>Value</b>
Typography	Monospace
Colors	Default: White text on black background Notifications: Red text on black background
Icon	Regular

Figure 4.2: User interface

and Tcpdump.

**Communications Interfaces** To send an alert to the alert, the user would be using the email as method of communication. Hence, the host “smtp.gmail.com” and port 465 (standard TCP) would be in use.

# Chapter 5

## Conclusion

In this effort, an active method to identify ARP spoofing was developed. It was demonstrated that the method employed here is more quicker and more adaptable than passive detection methods. The time lag between learning new addresses and spotting spoofing is as little as possible since active approach is employed here to investigate the authenticity of ARP traffic on a per packet basis. Despite the numerous fixes offered, brand-new attack methods could nonetheless result in brand-new security issues because the ARP protocol has a few fundamental flaws. Therefore, more research is needed to address the ARP protocol's underlying security flaws. This will show us the ARP traffic and how the attack is happening and by the scripts which the alert notification send to the administrator, this can prevent the ARP spoofing attack from happening. The user will receive an alert by telling the system is under ARP spoofing attack and they terminate the particular session by disconnecting the connection

## 5.1 Future Works

The addition of newer, more modern, and more sophisticated features to this project will be made possible by future research, studies, and advancements in the field of ARP spoofing. The tool can be used in a variety of circumstances. Although challenging, designing a GUI is not impossible.

# References

1. Sherin Hijazi, Mohammad S. Obaidat. A New Detection and Prevention System for ARP Attacks Using Static Entry , IEEE Systems Journal Volume: 13, Issue: 3 , Sept 2019
2. Mohammad Z. Masoud, Yousf Jaradat and Ismael Jannoud , On Preventing ARP poisoning attack utilizing software defined network (SDN) paradigm , Nov 2015
3. <https://linuxhint.com/wireshark-command-line-interface-tshark/>
4. <https://www.hindawi.com/journals/scn/2022/2196998/>
5. Mahesh V. Tripunitara, Partha Dutta. A Middleware Approach to Asynchronous and Backward Compatible Detection and Prevention of ARP Cache Poisoning
6. Sumit Kumar, Shashikala Tapaswi. A Centralized Detection and Prevention Technique against ARP Poisoning , : July 2012
7. <https://www.researchgate.net/publication/329078115> A New Detection and Prevention System for ARP Attacks Using Static Entry



8. Haider Salim; Zhitang Li, Hao Tu, Zhengbiao Guo. Preventing ARP Spoofing Attacks through Gratuitous Decision Packet, IEEE, Oct 2012
9. Seung Yeob Nam, Dongwon Kim, Jeongeun KimEnhanced. ARP: Preventing ARP Poisoning-Based Man-in-the-Middle Attacks , IEEE Communications Letters (Volume: 14 , Issue: 2 , February 2010)
10. Daesung Moon , Jae Dong Lee, Young-Sik Jeong ,Jong Hyuk Park . RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks , The Journal of Supercomputing volume 72, pages1740–1756(2016) , Dec 2014
11. Nikhil Tripathi, BM Mehtre, Analysis of Various ARP Poisoning Mitigation Techniques: A comparison July 2014
12. Seung Yeob Nam ,Sirojiddin Djuraev , Minho Park . Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks , Elsevier – Computer networks Volume 57 Dec 2013
13. Seung Yeob Nam , Sirojiddin Jurayev, Seung-Sik Kim, Kwonhue Choi and Gyu Sang Choi ,Mitigating ARP poisoning-based man-in-themiddle attacks in wired or wireless LAN , EURASIP Journal on Wireless Communications and Networking volume 2012
14. A Holistic Approach to ARP Poisoning and Countermeasures by Using PracticalExamples and Paradigm, Faisal Rahman, P. Kamal Computer Science 2014

15. A New Detection and Prevention System for ARP Attacks Using Static Entry

Sherin Hijazi, M. Obaidat Computer Science IEEE Systems Journal 2019